Guia 9 de lenguajes: Resultados basicos presentados en paradigma recursivo

November 3, 2024

Nota: Los ejercicios que tienen (S) son para una "Segunda vuelta" es decir conviene hacerlos una vez que ya se completó la guía haciendo los otros y ya se tiene mas madurez e intuición basica sobre los conceptos. Los que tienen (O) son opcionales por lo cual no se toman en los examenes.

En esta guia presentaremos varios resultados basicos de computabilidad, expresados en el paradigma recursivo, ya que es el mas habitual y comodo. Varios de estos resultados pueden ser establecidos y probados en forma natural dentro del paradigma de la computabilidad efectiva (ver apunte). A ellos los enunciaremos dentro del paradigma de Godel y los probaremos rigurosamente usando la teoria desarrollada hasta ahora. Sin envargo, veremos que hay otros resultados que son dependientes del desarrollo matematico hecho y aportan nueva informacion al paradigma filosofico (la indecidibilidad del halting problem, por ejemplo).

Como veremos muchas de las pruebas seran de naturaleza imperativa basadas en la equivalencia del paradigma de Godel con el imperativo de Neumann.

Lema de division por casos para funciones Σ-recursivas

Usando los resultados probados en las dos anteriores batallas entre los paradigmas de Godel y Neumann podemos probar el siguiente resultado el cual a priori no parese facil de probar si nos quedamos solo en el contexto del paradigma Godeliano.

Lemma 1 *Supongamos $f_i : D_{f_i} \subseteq \omega^n \times \Sigma^{*m} \to O$, $i = 1, ..., k$, son funciones Σ-recursivas tales que $D_{f_i} \cap D_{f_j} = \emptyset$ para $i \neq j$. Entonces la funcion $f_1 \cup ... \cup f_k$ es Σ-recursiva.*

Proof. Probaremos el caso $k = 2$ y $O = \Sigma^*$. Ademas supondremos que $n = m = 1$. Sean $\mathcal{P}_1$ y $\mathcal{P}_2$ programas que computen las funciones f_1 y f_2, respectivamente. Para $i = 1, 2$, definamos

$$H_i = \lambda t x_1 \alpha_1 \left[Halt^{1,1}(t, x_1, \alpha_1, \mathcal{P}_i) \right]$$

Notar que $D_{H_i} = \omega^2 \times \Sigma^*$ y que H_i es Σ-mixta. Ademas sabemos que la funcion $Halt^{1,1}$ es $(\Sigma \cup \Sigma_p)$-p.r. por lo cual resulta facilmente que H_i es $(\Sigma \cup \Sigma_p)$-p.r.. Por el Teorema de Independencia del Alfabeto tenemos que H_i es Σ-p.r.. Entonces H_i es Σ-computable por lo cual tenemos que hay un macro:

$$[\text{IF } H_i(\text{V1}, \text{V2}, \text{W1}) \text{ GOTO A1}]$$

Para hacer mas intuitivo el uso de este macro lo escribiremos de la siguiente manera

$$\big[\text{IF } Halt^{1,1}(\text{V1}, \text{V2}, \text{W1}, \mathcal{P}_i) \text{ GOTO A1}\big]$$

Ya que cada f_i es Σ-recursiva, hay macros

$$[\text{W2} \leftarrow f_1(\text{V1}, \text{W1})]$$
$$[\text{W2} \leftarrow f_2(\text{V1}, \text{W1})]$$

Sea $\mathcal{P}$ el siguiente programa:

$$\begin{array}{ll} \text{L1} & \text{N20} \leftarrow \text{N20} + 1 \\ & \big[\text{IF } Halt^{1,1}(\text{N20}, \text{N1}, \text{P1}, \mathcal{P}_1) \text{ GOTO L2}\big] \\ & \big[\text{IF } Halt^{1,1}(\text{N20}, \text{N1}, \text{P1}, \mathcal{P}_2) \text{ GOTO L3}\big] \\ & \text{GOTO L1} \\ \text{L2} & [\text{P1} \leftarrow f_1(\text{N1}, \text{P1})] \\ & \text{GOTO L4} \\ \text{L3} & [\text{P1} \leftarrow f_2(\text{N1}, \text{P1})] \\ \text{L4} & \text{SKIP} \end{array}$$

Notese que $\mathcal{P}$ computa la funcion $f_1 \cup f_2$ ■

Ejercicio 1: Si $P : D_P \subseteq \omega \times \omega^n \times \Sigma^{*m} \to \omega$ es un predicado Σ-recursivo y D_P es Σ-recursivo, entonces la funcion $M(P)$ es Σ-recursiva.

Lema de restriccion de funciones Σ-recursivas

Nos sera util tambien el siguiente resultado.

Lemma 2 *Supongamos $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to O$ es Σ-recursiva y $S \subseteq D_f$ es Σ-r.e., entonces $f|_S$ es Σ-recursiva.*

Proof. Si $S = \emptyset$, entonces $f|_S = \emptyset$ y por lo tanto $f|_S$ es Σ-recursiva. Supongamos $S \neq \emptyset$. Haremos el caso $n = m = 1$ y $O = \Sigma^*$. Tenemos que hay una $F : \omega \to \omega \times \Sigma^*$ tal que $\operatorname{Im} F = S$ y $F_{(1)}$, $F_{(2)}$ son Σ-recursivas. Ya que f, $F_{(1)}$ y $F_{(2)}$ son Σ-computables, hay macros

$$[\text{W2} \leftarrow f(\text{V1}, \text{W1})]$$
$$\big[\text{V2} \leftarrow F_{(1)}(\text{V1})\big]$$
$$\big[\text{W1} \leftarrow F_{(2)}(\text{V1})\big]$$

Ya que los predicados $D = \lambda xy[x \neq y]$ y $D' = \lambda\alpha\beta[\alpha \neq \beta]$ son Σ-p.r., tenemos que son Σ-computables, por lo cual hay macros

$$[\text{IF } D(\text{V1}, \text{V2}) \text{ GOTO A1}]$$
$$[\text{IF } D'(\text{W1}, \text{W2}) \text{ GOTO A1}]$$

Para para hacer mas amigable la lectura los escribieremos de la siguiente manera

$$[\text{IF V1} \neq \text{V2 GOTO A1}]$$
$$[\text{IF W1} \neq \text{W2 GOTO A1}]$$

Sea $\mathcal{P}$ el siguiente programa

L2	$[\text{N2} \leftarrow F_{(1)}(\text{N20})]$
	$[\text{P2} \leftarrow F_{(2)}(\text{N20})]$
	$[\text{IF N1} \neq \text{N2 GOTO L1}]$
	$[\text{IF P1} \neq \text{P2 GOTO L1}]$
	$[\text{P1} \leftarrow f(\text{N1}, \text{P1})]$
	GOTO L3
L1	$\text{N20} \leftarrow \text{N20} + 1$
	GOTO L2
L3	SKIP

Es facil ver que $\mathcal{P}$ computa a $f|_S$ ■

Conjuntos Σ-r.e. y Σ-r.

Daremos primero algunas propiedades basicas de los conjuntos Σ-r.e. y Σ-r. El siguiente resultado puede probarse facilmente dentro del paradigma Godeliano y lo dejamos como ejercicio.

Ejercicio 1,5: Sea Σ un alfabeto finito. Supongamos $S_1, S_2 \subseteq \omega^n \times \Sigma^{*m}$ son conjuntos Σ-recursivos. Entonces $S_1 \cup S_2$, $S_1 \cap S_2$ y $S_1 - S_2$ son Σ-recursivos

Lemma 3 *Sea Σ un alfabeto finito. Se tiene que*

(1) Supongamos $S_1, S_2 \subseteq \omega^n \times \Sigma^{*m}$ son conjuntos Σ-r.e.. Entonces $S_1 \cup S_2$ es Σ-r.e.

(2) Supongamos $S_1, S_2 \subseteq \omega^n \times \Sigma^{*m}$ son conjuntos Σ-r.e.. Entonces $S_1 \cap S_2$ es Σ-r.e.

(3) Sea $S \subseteq \omega^n \times \Sigma^{*m}$. Si S es Σ-recursivo, entonces S es Σ-recursivamente enumerable

Proof. Los tres resultados en su version imperativa fueron probados o dejados como ejercicio en la Guia 8, por lo que podemos aplicar los teoremas que nos dicen que los paradigmas recursivo e imperativo son equivalentes. ■

Tal como veremos mas adelante hay conjuntos Σ-recursivamente enumerables los cuales no son Σ-recursivos. Sin envargo tenemos el siguiente interesante resultado.

Lemma 4 *Sea $S \subseteq \omega^n \times \Sigma^{*m}$. Si S y $(\omega^n \times \Sigma^{*m}) - S$ son Σ-recursivamente enumerables, entonces S es Σ-recursivo*

Proof. (Prueba cheta) Por definicion, para probar que S es Σ-recursivo deberemos probar que $\chi_S^{\omega^n \times \Sigma^{*m}}$ es Σ-recursiva. Notese que $\chi_S^{\omega^n \times \Sigma^{*m}} = C_1^{n,m}|_S \cup C_0^{n,m}|_{(\omega^n \times \Sigma^{*m}) - S}$. O sea que por el lema de division por casos, solo nos resta probar que $C_1^{n,m}|_S$ y $C_0^{n,m}|_{(\omega^n \times \Sigma^{*m}) - S}$ son Σ-recursivas. Pero esto se desprende directamente del Lema 2 ya que $C_1^{n,m}$ y $C_0^{n,m}$ son Σ-recursivas y por hipotesis S y $(\omega^n \times \Sigma^{*m}) - S$ son Σ-recursivamente enumerables. ■

Ejercicio 2: (Prueba intuitiva) Dar una prueba imperativa del lema anterior. Hint: inspirese en su analogo dentro del paradigma de la computabilidad efectiva, dado al final de la Guia 3.

El siguiente teorema es muy importante ya que caracteriza a los conjuntos Σ-r.e. Solo se tomara la prueba de la implicacion (2)⇒(3) aunque dejamos su prueba completa para el lector interesado.

Theorem 5 *Dado $S \subseteq \omega^n \times \Sigma^{*m}$, son equivalentes*

(1) S es Σ-recursivamente enumerable

(2) $S = I_F$, para alguna $F : D_F \subseteq \omega^k \times \Sigma^{*l} \to \omega^n \times \Sigma^{*m}$ tal que cada $F_{(i)}$ es Σ-recursiva.

(3) $S = D_f$, para alguna funcion Σ-recursiva f

(4) $S = \emptyset$ o $S = I_F$, para alguna $F : \omega \to \omega^n \times \Sigma^{*m}$ tal que cada $F_{(i)}$ es Σ-p.r.

Proof. El caso $n = m = 0$ es facil y es dejado al lector. Supongamos entonces que $n + m \geq 1$.

(2)⇒(3). Haremos el caso $k = l = 1$ y $n = m = 2$. El caso general es completamente analogo. Notese que entonces tenemos que $S \subseteq \omega^2 \times \Sigma^{*2}$ y $F : D_F \subseteq \omega \times \Sigma^* \to \omega^2 \times \Sigma^{*2}$ es tal que $\operatorname{Im} F = S$ y $F_{(1)}$, $F_{(2)}$, $F_{(3)}$, $F_{(4)}$ son Σ-recursivas. Para cada $i \in \{1, 2, 3, 4\}$, sea $\mathcal{P}_i$ un programa el cual computa a $F_{(i)}$. Sea $\leq$ un orden total sobre Σ. Definamos

$$H_i = \lambda t x_1 \alpha_1 \left[\neg Halt^{1,1}(t, x_1, \alpha_1, \mathcal{P}_i)\right]$$

Notar que $D_{H_i} = \omega^2 \times \Sigma^*$ y que H_i es Σ-mixta. Ademas sabemos que la funcion $Halt^{1,1}$ es $(\Sigma \cup \Sigma_p)$-p.r. por lo cual resulta facilmente que H_i es $(\Sigma \cup \Sigma_p)$-p.r.. Por la Proposicion de Independencia del Alfabeto tenemos que H_i es Σ-p.r.. Entonces H_i es Σ-computable por lo cual tenemos que hay un macro:

$$[\text{IF } H_i(\text{V2}, \text{V1}, \text{W1}) \text{ GOTO A1}]$$

Para hacer mas intuitivo el uso de este macro lo escribiremos de la siguiente manera

$$\left[\text{IF } \neg Halt^{1,1}(\text{V2}, \text{V1}, \text{W1}, \mathcal{P}_i) \text{ GOTO A1}\right]$$

Para $i = 1, 2$, definamos

$$E_i = \lambda x t x_1 \alpha_1 \left[x \neq E^{1,1}_{\#1}(t, x_1, \alpha_1, \mathcal{P}_i)\right]$$

Para $i = 3, 4$, definamos

$$E_i = \lambda t x_1 \alpha_1 \alpha \left[\alpha \neq E^{1,1}_{*1}(t, x_1, \alpha_1, \mathcal{P}_i)\right]$$

Dejamos al lector probar que las funciones E_i son Σ-p.r.. O sea que son Σ-computables por lo cual para cada $i \in \{1, 2\}$ hay un macro

$$[\text{IF } E_i(\text{V2}, \text{V3}, \text{V1}, \text{W1}) \text{ GOTO A1}]$$

y para cada $i \in \{3, 4\}$ hay un macro

$$[\text{IF } E_i(\text{V2}, \text{V1}, \text{W1}, \text{W2}) \text{ GOTO A1}]$$

Haremos mas intuitiva la forma de escribir estos macros, por ejemplo para $i = 1$, lo escribiremos de la siguiente manera

$$\left[\text{IF V2} \neq E^{1,1}_{\#1}(\text{V3}, \text{V1}, \text{W1}, \mathcal{P}_1) \text{ GOTO A1}\right]$$

Ya que la funcion $f = \lambda x[(x)_1]$ es Σ-p.r., ella es Σ-computable por lo cual hay un macro

$$[\text{V2} \leftarrow f(\text{V1})]$$

el cual escribiremos de la siguiente manera:

$$[\text{V2} \leftarrow (\text{V1})_1]$$

Similarmente hay macros:

$$[\text{W1} \leftarrow *^{\leq}(\text{V1})_3]$$

$$[\text{V2} \leftarrow (\text{V1})_2]$$

(dejamos al lector entender bien el funcionamiento de estos macros). Sea $\mathcal{P}$ el siguiente programa de $\mathcal{S}^{\Sigma}$:

$$\begin{array}{l}
\text{L1 N20} \leftarrow \text{N20} + 1 \\
[\text{N10} \leftarrow (\text{N20})_1] \\
[\text{N3} \leftarrow (\text{N20})_2] \\
[\text{P3} \leftarrow *^{\leq}(\text{N20})_3] \\
\left[\text{IF } \neg Halt^{1,1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_1) \text{ GOTO L1}\right] \\
\left[\text{IF } \neg Halt^{1,1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_2) \text{ GOTO L1}\right] \\
\left[\text{IF } \neg Halt^{1,1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_3) \text{ GOTO L1}\right] \\
\left[\text{IF } \neg Halt^{1,1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_4) \text{ GOTO L1}\right] \\
\left[\text{IF N1} \neq E^{1,1}_{\#1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_1) \text{ GOTO L1}\right] \\
\left[\text{IF N2} \neq E^{1,1}_{\#1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_2) \text{ GOTO L1}\right] \\
\left[\text{IF P1} \neq E^{1,1}_{*1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_3) \text{ GOTO L1}\right] \\
\left[\text{IF P2} \neq E^{1,1}_{*1}(\text{N10}, \text{N3}, \text{P3}, \mathcal{P}_4) \text{ GOTO L1}\right]
\end{array}$$

Dejamos al lector la tarea de comprender el funcionamiento de este programa y convenserse de que computa la funcion $p_1^{2,2}|_S$. Pero entonces $p_1^{2,2}|_S$ es Σ-computable por lo cual es Σ-recursiva, lo cual prueba (3) ya que $Dom(p_1^{2,2}|_S) = S$.

(3)$\Rightarrow$(4). Supongamos $S \neq \emptyset$. Sea $(z_1, ..., z_n, \gamma_1, ..., \gamma_m) \in S$ fijo. Sea $\mathcal{P}$ un programa el cual compute a f y Sea $\leq$ un orden total sobre Σ. Sea $P : \mathbf{N} \to \omega$ dado por $P(x) = 1$ sii

$$Halt^{n,m}\left((x)_{n+m+1}, (x)_1, ..., (x)_n, *^{\leq}((x)_{n+1}), ..., *^{\leq}((x)_{n+m})), \mathcal{P}\right) = 1$$

Es facil ver que P es $(\Sigma \cup \Sigma_p)$-p.r. por lo cual es Σ-p.r.. Sea $\bar{P} = P \cup C_0^{1,0}|_{\{0\}}$. Para $i = 1, ..., n$, definamos $F_i : \omega \to \omega$ de la siguiente manera

$$F_i(x) = \begin{cases} (x)_i & \text{si} \quad \bar{P}(x) = 1 \\ z_i & \text{si} \quad \bar{P}(x) \neq 1 \end{cases}$$

Para $i = n+1, ..., n+m$, definamos $F_i : \omega \to \Sigma^*$ de la siguiente manera

$$F_i(x) = \begin{cases} *^{\leq}((x)_i) & \text{si} \quad \bar{P}(x) = 1 \\ \gamma_{i-n} & \text{si} \quad \bar{P}(x) \neq 1 \end{cases}$$

Por el lema de division por casos (para funciones Σ-p.r.), cada F_i es Σ-p.r.. Es facil ver que $F = [F_1, ..., F_{n+m}]$ cumple (4). ■

El halting problem y los conjuntos A y N

Cuando $\Sigma \supseteq \Sigma_p$, podemos definir

$$AutoHalt^{\Sigma} = \lambda\mathcal{P}\left[(\exists t \in \omega)\ Halt^{0,1}(t, \mathcal{P}, \mathcal{P})\right].$$

Notar que el dominio de $AutoHalt^{\Sigma}$ es Pro^{Σ} y que para cada $\mathcal{P} \in \text{Pro}^{\Sigma}$ tenemos que

(*) $AutoHalt(\mathcal{P}) = 1$ sii $\mathcal{P}$ se detiene partiendo del estado $\|\mathcal{P}\|$.

Lemma 6 *Supongamos $\Sigma \supseteq \Sigma_p$. Entonces $AutoHalt^{\Sigma}$ no es Σ-recursivo.*

Proof. Supongamos $AutoHalt^{\Sigma}$ es Σ-recursivo y por lo tanto Σ-computable. Por la proposicion de existencia de macros tenemos que hay un macro

$$\big[\text{IF } AutoHalt^{\Sigma}(\text{W1}) \text{ GOTO A1}\big]$$

Sea $\mathcal{P}_0$ el siguiente programa de $\mathcal{S}^{\Sigma}$

$$\text{L1 } \big[\text{IF } AutoHalt^{\Sigma}(\text{P1}) \text{ GOTO L1}\big]$$

Note que

- $\mathcal{P}_0$ termina partiendo desde $\|\mathcal{P}_0\|$ sii $AutoHalt^{\Sigma}(\mathcal{P}_0) = 0$,

lo cual produce una contradiccion si tomamos en (*) $\mathcal{P} = \mathcal{P}_0$. ■

Usando el lema anterior y la Tesis de Church podemos probar el siguiente impactante resultado.

Theorem 7 *Supongamos $\Sigma \supseteq \Sigma_p$. Entonces $AutoHalt^{\Sigma}$ no es Σ-efectivamente computable. Es decir no hay ningun procedimiento efectivo que decida si un programa de $\mathcal{S}^{\Sigma}$ termina partiendo de si mismo.*

Proof. Si $AutoHalt^{\Sigma}$ fuera Σ-efectivamente computable, la Tesis de Church nos diria que es Σ-recursivo, contradiciendo el lema anterior. ■

Notese que $AutoHalt^{\Sigma}$ provee de un ejemplo natural en el cual la cuantificacion (no acotada) de un predicado Σ-p.r. con dominio rectangular no es Σ-efectivamente computable

Ahora estamos en condiciones de dar un ejemplo natural de un conjunto A que es Σ-recursivamente enumerable pero el cual no es Σ-recursivo.

Lemma 8 *Supongamos que $\Sigma \supseteq \Sigma_p$. Entonces*

$$A = \big\{\mathcal{P} \in \text{Pro}^{\Sigma} : AutoHalt^{\Sigma}(\mathcal{P}) = 1\big\}$$

es Σ-r.e. y no es Σ-recursivo. Mas aun el conjunto

$$N = \big\{\mathcal{P} \in \text{Pro}^{\Sigma} : AutoHalt^{\Sigma}(\mathcal{P}) = 0\big\}$$

no es Σ-r.e.

Proof. Para ver que A es Σ-r.e. se lo puede hacer imperativamente dando un programa (usando macros) que enumere a A. De esta forma probariamos que A es Σ-enumerable y por lo tanto es Σ-r.e.. Daremos ahora una prueba no imperativa de este hecho, es decir mas propia del paradigma funcional. Sea $P = \lambda t\mathcal{P}\left[Halt^{0,1}(t, \mathcal{P}, \mathcal{P})\right]$. Note que P es Σ-p.r. por lo que $M(P)$ es Σ-r.. Ademas note que $D_{M(P)} = A$, lo cual implica que A es Σ-r.e..

Supongamos ahora que N es Σ-r.e.. Entonces la funcion $C_0^{0,1}|_N$ es Σ-recursiva ya que $C_0^{0,1}$ lo es. Ademas ya que A es Σ-r.e. tenemos que $C_1^{0,1}|_A$ es Σ-recursiva. Ya que

$$AutoHalt^{\Sigma} = C_1^{0,1}|_A \cup C_0^{0,1}|_N$$

el lema de division por casos nos dice que $AutoHalt^{\Sigma}$ es Σ-recursivo, contradiciendo el Lema 6. Esto prueba que N no es Σ-r.e..

Finalmente supongamos A es Σ-recursivo. Entonces el conjunto

$$N = (\Sigma^* - A) \cap \mathrm{Pro}^{\Sigma}$$

deberia serlo, lo cual es absurdo. Hemos probado entonces que A no es Σ-recursivo. ■

Usando la Tesis de Church obtenemos el siguiente resultado.

Proposition 9 *Supongamos que $\Sigma \supseteq \Sigma_p$. Entonces A es Σ-efectivamente enumerable y no es Σ-efectivamente computable. El conjunto N no es Σ-efectivamente enumerable. Es decir, A puede ser enumerado por un procedimiento efectivo pero no hay ningun procedimiento efectivo que decida la pertenencia a A y no hay ningun procedimiento efectivo que enumere a N. Mas aun, si un procedimiento efectivo da como salida siempre elementos de N, entonces hay una cantidad infinita de elementos de N los cuales nunca da como salida*

Ejercicio 3: (S) Justifique la ultima aseveracion en la proposicion anterior

Cabe destacar aqui que el dominio de una funcion Σ-recursiva no siempre sera un conjunto Σ-recursivo. Por ejemplo si tomamos Σ tal que $\Sigma \supseteq \Sigma_p$, entonces $C_1^{0,1}|_A$ es una funcion Σ-recursiva ya que es la restriccion de la funcion Σ-recursiva $C_1^{0,1}$ al conjunto Σ-r.e. A, pero $\mathrm{Dom}(C_1^{0,1}|_A) = A$ no es Σ-recursivo.

Ejercicio 4: Pruebe que no es cierta la siguiente (hermosa y tentadora) propiedad

- Si $f : S \subseteq \omega^n \times \Sigma^{*m} \to \omega$ es una funcion Σ-computable, entonces hay un macro

$$\left[\text{IF } \chi_S^{\omega^n \times \Sigma^{*m}}(\text{V1}, ..., \text{V}\bar{n}, \text{W1}, ..., \text{W}\bar{m}) \text{ GOTO A1}\right]$$

Ejercicio 5: V o F o I. justifique.

(a) Sea Σ un alfabeto y sean $n, m \in \omega$. Entonces el dominio de $T^{n,m}$ es rectangular

(b) Para cada $\mathcal{P} \in \text{Pro}^{\Sigma}$ hay un $\mathcal{P}' \in \text{Pro}^{\Sigma}$ tal que $\text{Dom}(\Psi^{1,0,\#}_{\mathcal{P}'}) = \omega - \text{Dom}(\Psi^{1,0,\#}_{\mathcal{P}})$.

(c) Sea $\mathcal{P} \in \text{Pro}^{\Sigma}$ y supongamos que para cada $x \in \omega$, $\mathcal{P}$ termina partiendo de $((x, 0, 0...), (\varepsilon, \varepsilon, ...))$ en a lo sumo $n(\mathcal{P})^2 + x$ pasos. Entonces $\Psi^{1,0,*}_{\mathcal{P}}$ es Σ-p.r.

(d) Hay $\mathcal{P} \in \text{Pro}^{\Sigma}$ tal que $0 \in D_{\Psi^{1,0,\#}_{\mathcal{P}}}$ y

$$\Psi^{1,0,\#}_{\mathcal{P}}(0) = 1 + \min_t(i^{1,0}(t, 0, \mathcal{P}) = n(\mathcal{P}) + 1)$$

Ejercicio 6: (Opcional) (S) Pruebe que la reciproca del Ejercicio 1 no es cierta, es decir de un predicado Σ-recursivo $P : D_P \subseteq \omega \times \omega^n \times \Sigma^{*m} \to \omega$ el cual cumpla que $M(P)$ es Σ-recursiva y que D_P no es Σ-recursivo (Hint: tome $P = C^{1,1}_1|_{\omega \times A}$)

Con los resultados anteriores estamos en condiciones de dar un ejemplo de un predicado Σ-recursivo, cuya minimizacion no es Σ-efectivamente computable (y por lo tanto no es Σ-recursiva). Aceptaremos el resultado sin demostracion.

Proposition 10 *Supongamos que $\Sigma \supseteq \Sigma_p$. Sea $P = C^{0,1}_1|_A \circ \lambda t \alpha \left[\alpha^{1 \dot{-} t} \text{SKIP}^t\right] |_{\omega \times \text{Pro}^{\Sigma}}$. El predicado P es Σ-recursivo pero la funcion $M(P)$ no es Σ-efectivamente computable (y por lo tanto no es Σ-recursiva)*

Ejercicio 7: (Opcional) (S) Daremos aqui una guia para probar la proposicion anterior.

(a) Pruebe que P es Σ-recursivo

(b) Pruebe que $D_{M(P)} = \text{Pro}^{\Sigma}$

(c) Para cada $\mathcal{P} \in \text{Pro}^{\Sigma}$ se tiene que

$$M(P)(\mathcal{P}) = 0 \text{ sii } \mathcal{P} \in A$$

(d) Pruebe que $AutoHalt^{\Sigma} = \lambda x[x = 0] \circ M(P)$ por lo cual $M(P)$ no es Σ-recursiva

(e) $M(P)$ no es Σ-efectivamente computable